

© EPODOC / EPO

PN - JP4332045 A 19921119
PD - 1992-11-19
PR - JP19910101467 19910507
OPD - 1991-05-07
TI - ARITHMETIC PROCESSOR
IN - SAITO YUTAKA
PA - FUJI ELECTRIC CO LTD
IC - G06F9/46

© PAJ / JPO

PN - JP4332045 A 19921119
PD - 1992-11-19
AP - JP19910101467 19910507
IN - SAITO YUTAKA
PA - FUJI ELECTRIC CO LTD
TI - ARITHMETIC PROCESSOR
AB - PURPOSE: To optionally set up the restarting position of an execution instruction after ending an interruption program by storing an instruction for changing the storage value of a stack register in the interruption program and executing the stored instruction by an instruction decoder and a writing circuit.
- CONSTITUTION: A program instruction read out from a program memory 20 is stored in a latch circuit 12 and decoded by an instruction decoder 11. When the instruction is a storage value changing instruction for the stack register 16, the decoder 11 generates an address change instruction detecting signal. At the time of inputting an interruption signal, a timing signal generating circuit 13 stores and outputs an interruption detection signal. At the time of inputting a write command signal, the circuit 13 releases the storing state of the interruption detection signal. At the time of detecting an address change instruction in an interruption program, a writing circuit 15 writes an address value included in the address change instruction in the register 16.
I - G06F9/46

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平4-332045

(43) 公開日 平成4年(1992)11月19日

(51) Int.Cl.³

G 0 6 F 9/46

識別記号

3 1 0 Z 8120-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数1(全 6 頁)

(21) 出願番号 特願平3-101467

(22) 出願日 平成3年(1991)5月7日

(71) 出願人 000005234

富士電機株式会社

神奈川県川崎市川崎区田辺新田1番1号

(72) 発明者 斉藤 豊

神奈川県川崎市川崎区田辺新田1番1号

富士電機株式会社内

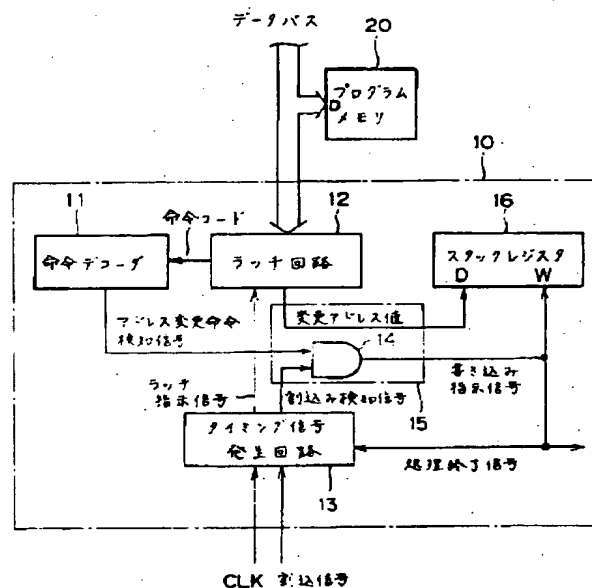
(74) 代理人 弁理士 谷 義一 (外1名)

(54) 【発明の名称】 演算処理装置

(57) 【要約】

【目的】 割込みプログラム終了後の実行命令の再開位置を任意に設定する。

【構成】 割込みプログラムの中にスタックレジスタ16の格納値を変更する命令を記載し、この命令を命令デコード11および書き込み回路15で実行することにより実行再開のプログラム命令の位置を変更する。



【特許請求の範囲】

【請求項1】 第1のプログラムをプログラム命令単位で実行中に割込信号を入力した場合は、当該第1のプログラムにおいて実行しようとしていたプログラム命令の位置情報を記憶手段に記憶し、次に、割込用に定めた第2のプログラムを実行した後、前記記憶手段の位置情報の示すプログラム命令から実行を再開する演算処理装置において、前記記憶手段に記憶の前記位置情報を変更する旨を示す命令コードと、変更すべき新たなプログラム命令の位置情報とを含む変更命令を予め定め、実行対象のプログラム命令が前記変更命令であることを検出する命令識別手段と、該命令識別手段により前記変更命令が検出されたときには、当該変更命令に含まれる位置情報を前記記憶手段に更新的に書き込むことにより、前記第2プログラムの終了後に実行すべきプログラム命令を変更する変更手段とを具えたことを特徴とする演算処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、プログラムをプログラム命令単位で演算実行する演算処理装置に関する。

【0002】

【従来の技術】 中央演算処理装置（CPU）は内蔵または外付のシステムプログラムメモリに格納された演算プログラムをプログラム命令単位で読出して、演算実行する。

【0003】 通常、CPUはシステムプログラムメモリの開始アドレスをアドレスカウンタに設定して、先頭のプログラム命令を読出した後、プログラム命令の実行毎に、アドレスカウンタを“1”ずつ更新し、実行対象のプログラム命令を順次に読出す。

【0004】 また、CPUの割込み端子に割込み信号を入力したときは、CPUは現在実行しているプログラムを一時中断し、割込信号に対応した別の割込み用プログラムを実行する。このプログラムを実行した後、CPUは中断していた元のプログラムの実行を再開することができる。割込み処理を実行する場合、CPU内の具体的なシステム処理は次のように行われる。

【0005】 割込み信号を入力すると、CPUは、現在、アドレスカウンタに設定されているプログラム命令の読出しアドレスを割込みプログラム終了後の復帰先アドレスとして、スタックレジスタと呼ばれるCPU内の専用レジスタに格納する。

【0006】 この後、CPUはアドレスカウンタに割込みプログラムの開始アドレスを設定し、アドレスカウンタの指示するシステムプログラム内のアドレスから割込み用プログラムを順次に読出し、実行する。割込み用プログラムの最後尾に記載された復帰命令（リターン命令）をCPUが読出すと、CPUはスタックレジスタに

格納されたアドレス値をアドレスカウンタに初期設定し、このアドレスカウンタの示すアドレスからプログラム命令の読出し演算実行を再開する。

【0007】 このような割込処理を起動する割込み信号はCPUに対して複数点入力可能であり、複数の割込用プログラムを実行することが可能となっている。

【0008】

【発明が解決しようとする課題】 しかしながら、従来この種、演算処理装置では割込み信号と割込用プログラムが1対1に対応しているため、多数の割込み処理を実行しなければならない場合は複数点の割込み信号を必要とする。

【0009】 そこで、本発明の目的は、上述の点に鑑みて、割込み信号の点数に関係なく任意の本数の割込み用処理プログラムを実行させるのみならず、従来にはない新規の割込み処理を達成することの可能な演算処理装置を提供することにある。

【0010】

【課題を解決するための手段】 このような目的を達成するために、本発明は、第1のプログラムをプログラム命令単位で実行中に割込信号を入力した場合は、当該第1のプログラムにおいて実行しようとしていたプログラム命令の位置情報を記憶手段に記憶し、次に、割込用に定めた第2のプログラムを実行した後、前記記憶手段の位置情報の示すプログラム命令から実行を再開する演算処理装置において、前記記憶手段に記憶の前記位置情報を変更する旨を示す命令コードと、変更すべき新たなプログラム命令の位置情報とを含む変更命令を予め定め、実行対象のプログラム命令が前記変更命令であることを検出する命令識別手段と、該命令識別手段により前記変更命令が検出されたときには、当該変更命令に含まれる位置情報を前記記憶手段に更新的に書き込むことにより、前記第2プログラムの終了後に実行すべきプログラム命令を変更する変更手段とを具えたことを特徴とする。

【0011】

【作用】 本発明は、スタックレジスタと呼ばれる記憶手段に格納されたプログラム命令の開始位置に基づいて、割込み用第2プログラム終了後、再開すべきプログラム命令の位置が決定されることに着目し、割込み用第2プログラムの中に記載した変更命令を検出し、記憶手段の位置情報を変更することにより、たとえば他の割込みプログラムの先頭位置に直接の実行手順を移行させる。

【0012】

【実施例】 以下、図面を参照して本発明実施例を詳細に説明する。

【0013】 図1は本発明第1実施例の回路構成例を示す。

【0014】 図1において、10は中央演算処理装置（CPU）を示す。

【0015】 本実施例ではCPU10中の従来から周知

3

の回路についての説明を省略し、本発明に関わる回路構成のみ説明することにする。

【0016】不図示のアドレスカウンタのアドレス指定によりプログラムメモリ20から読出されたプログラム命令はデータバスに送出され、ラッチ回路12に保持(ラッチ)される。ラッチ回路12に対するラッチ指示信号はタイミング信号発生回路13においてマシクロックに基づき作成される。

【0017】ラッチ回路12に保持されたコード信号形態のプログラム命令は命令デコーダ(命令コード解読器)11に入力され、命令デコーダにおいて命令内容の解説が行われる。

【0018】命令デコーダ(本発明の命令識別手段)11は従来の命令についての解説を行う他、読出しの命令コードが、本発明のスタックレジスタ16に対する格納値変更命令(以下、アドレス変更命令と称す)を示す特定コードであることをコード比較により検出した場合、アドレス変更命令検知信号を発生する。

【0019】タイミング信号発生回路13はマシクロックCLKを入力し、ラッチ回路12に対してラッチ指示信号を出力すると共に、割込信号を入力したときは割込み検知信号を保持出力する。また、タイミング信号発生回路13は、後述の書き込み指示信号を入力したときは割込み検知信号の保持出力状態を解除する。

【0020】書き込み回路15は、割込みプログラム中のアドレス変更命令が検出されたときに、このアドレス変更命令に含まれるアドレス値をスタックレジスタ(本発明の記憶手段)16に書き込む。本実施例ではタイミング信号発生回路13に保持された割込検知信号と命令デコーダ11のアドレス変更命令検知信号が同時に発生したことをアンドゲート14により検出し、スタックレジスタ16に対する書き込み指示信号を発生する。

【0021】このような回路構成において実行する割込み処理手順を図2のフローチャートを参照して説明する。図2のフローチャートは図1のプログラムメモリ20に格納された制御手順を示す。この制御手順はメインプログラムと割込みプログラムから構成されている。メインプログラムはたとえば、一定周期で起動され、21個のプログラム命令で構成されている。このメインプログラムはプログラムメモリ20のアドレス“0”～アドレス“20”(10進数)に格納されているものとする。

【0022】割込みプログラムは割込信号により起動され、4個のプログラム命令で構成されている。割込みプログラムはプログラムメモリ20のアドレス“100”～“103”(10進数)に格納されているものとする。

【0023】また、割込みプログラムの第2番目のプログラム命令(アドレス101)が本発明のアドレス変更命令となっている。このアドレス変更命令は命令の種類

4

を示す命令コードと、変更すべきアドレス値から構成される。本実施例では説明のためにメインプログラムの終了命令(図2参照)の格納されたアドレス値“20”がアドレス変更命令の中に含まれているものとする。

【0024】図1に戻り、回路動作を説明する。

【0025】電源起動に応じて、CPU10内のアドレスカウンタにはプログラムメモリ20の先頭アドレスと同じ値“0”が初期設定される。次にタイミング信号発生回路13の発生する同期信号に従ってアドレスカウンタの計数値が不図示のアドレスバスを介してプログラムメモリ20に転送され、プログラムメモリ20のアドレス“0”からプログラム命令A1(図2参照)が読出される。読出されたプログラム命令はデータバスを介してラッチ回路12に保持される。

【0026】命令デコーダ11はラッチ回路12に保持されたプログラム命令A1の中の命令コード内容を解説し、その命令に対応する検知信号をオンする。この検知信号に応じてその命令を実行する論理回路、たとえばCPU内のレジスタの書き込み回路、読出し回路、演算回路等がその命令の示す情報処理を実行する。

【0027】論理回路の処理終了信号に応じてアドレスカウンタが“1”に更新され、プログラムメモリ20のアドレス“1”から、次ステップのプログラム命令A2が読出され、ラッチ回路12に保持された後、上述と同様の処理手順でプログラム命令A2の示す情報処理が実行される。また、タイミング信号発生回路13はオンの割込み検知信号をアンドゲート14に保持出力する。

【0028】このようにして、アドレス“10”のプログラム命令A3を実行しようとするときに割込み信号がCPU10に入力されると、従来周知の処理手順により、アドレスカウンタの計数値“10”がスタックレジスタ16に格納され、次にCPU10内のレジスタに格納された、割込プログラムの先頭アドレス値“100”がアドレスカウンタに設定される。

【0029】この結果、CPU10の実行命令として図2に示すように、プログラムメモリ20のアドレス“100”からプログラム命令B1が読出される。

【0030】このプログラム命令の実行終了の後、アドレスカウンタが“101”に更新されると、アドレス変更命令がプログラムメモリ20から読出される。

【0031】ラッチ回路12に保持されたこのアドレス変更命令が命令デコーダ11により解説されると、アドレス変更命令検知信号がオンとなる。この結果、アンドゲート14の2入力と共にオンとなるのでゲートが開き、書き込み指示信号がオンレベルとなる。

【0032】スタックレジスタ16はオンの書き込み指示信号により、ラッチ回路12のプログラム命令中の変更アドレス値を上書き記憶することによりアドレス変更を行う。

【0033】以下、CPU10はアドレスカウンタを実

5

行して順次に割込プログラムのプログラム命令を実行する。

【0034】割込プログラムの最後尾に位置する復帰命令をCPU10が読出し、命令デコーダ11によりその命令内を解釈すると、従来処理同様、アドレスカウンタの設定値がスタックレジスタ16の格納値“20”に更新され、プログラムメモリ20のアドレス“20”から終了命令(図2参照)が読出される。この結果、CPU10はメインプログラムを終了し、命令待機状態となる。

【0035】以上、説明した例は、割込み信号によりメインプログラムを中断し、割込みプログラムを実行した後、メインプログラムを終了してしまう例であるが、アドレス変更命令の変更値を変えるのみで種々のプログラムを実行することができる。この実行形態を図3、図4に示す。

【0036】図3は、第1プログラムの実行中に割込信号を受けた場合、割込プログラムの終了後、第2プログラムを先頭プログラム命令から実行する例および割込プログラムの終了後、第3プログラムの途中のプログラム命令から実行する例を示す。

【0037】図4はループ形態で構成された第4プログラムから割込み信号により脱却し、割込プログラムを終了後別の第5プログラムにCPU10の実行手順を移行させる例を示す。

【0038】このような形態でプログラムを実行させることは復帰命令で実行手順を元に戻してしまう従来例ではできなかったことである。

【0039】以上、説明した第1実施例はプログラムメモリにリードオンリメモリ(ROM)を用い、プログラム命令がマシン語と呼ばれるCPUを作動させるプログラム言語で記載された演算処理装置に好適な例である。次に、実行対象のプログラムがベシック、フォートラン、その他高級言語や中間言語と呼ばれる言語で登録記憶される演算処理装置に好適な第2実施例を図5を用いて説明する。

【0040】図5において、CPU100、システムプログラムメモリ120、アプリケーションプログラムメモリ130、ワークメモリ140がバスに共通接続されている。

【0041】CPU100はマシン語形態のプログラム命令を実行するCPUチップであればよく、従来から知られているものを使用することができる。

【0042】システムプログラムメモリ120には装置制御たとえば、バス接続されたメモリに対して情報を読み書きする処理を実行するためのシステムプログラムと、中間言語のプログラム命令をマシン語のプログラム命令に翻訳するためのインタプリタと呼ばれるプログラムとが格納されている。

【0043】アプリケーションプログラムメモリ130

6

はユーザの作成した中間言語形態の演算対象の(アプリケーション)プログラムを格納する。アプリケーションプログラムは不図示のフロッピーディスク等からこのメモリにCPU100により転送される。

【0044】ワークメモリ140はCPU100の演算やシステム処理に用いるデータを格納する。本実施例では割込みプログラム終了後のCPU100の実行処理の戻り先を示すアドレス情報(スタックポインタと称す)はワークメモリ140内の専用領域に格納する。なおCPU100内のスタックレジスタは、システムプログラムを読出す際にアプリケーションプログラムの読出しアドレスを退避記憶させるために用いる。

【0045】このような構成において、CPU100はアプリケーションプログラムの先頭アドレスから中間言語のプログラム命令を読出すと、ワークメモリ140に一時記憶し、次に、インタプリタ121に従って、マシン語形態の命令に変換した後、変換後の命令を実行する。このとき、接続のメモリ140等に対して情報の読み書きを行う場合は、システムプログラム120を用いて命令に規定された内容を実行する。このような処理はインタプリタ方式の演算処理と呼ばれ、周知であるので詳細な説明は要しないであろう。

【0046】本実施例では、インタプリタ121の中に、スタックポインタの変更命令検出処理(図6のステップS100)およびスタックポインタの変更処理(図6のステップS110)を実行させるプログラム手順を記載しておく。

【0047】変更命令検出処理はより具体的にはアプリケーションプログラムメモリ130から読出した実行対象のプログラム命令(中間語)がスタックポインタの変更命令であるか否かをコード比較により判定する処理である。

【0048】スタックポインタ変更処理は、アプリケーションプログラムメモリ130から読出した上記スタックポインタの変更命令の中から変更値を抽出し、次に、ワークメモリ130のスタックポインタに変更値を書き込む処理をマシン語に翻訳する処理である。

【0049】このようなインタプリタを用いることによって、アプリケーションプログラム中の割込みプログラムに記載したスタックポインタの変更命令が検出されると、スタックポインタの変更処理内容がマシン語に翻訳され、CPU100により実行される。

【0050】この結果、アプリケーションプログラム中の割込プログラムを終了した後のCPU100の実行手順の戻り先はスタックポインタの変更値により決定される。

【0051】第1実施例はハードウェアにより本発明を実現する例であり、第2実施例はソフトウェアにより本発明を実現する例であるが、演算処理装置を用いる電子機器の仕様に応じて所望の形態を採用するとよい。

【発明の効果】以上、説明したように、本発明では、割

The block diagram illustrates the internal components and signal flow of a control system, likely for a semiconductor device. The main processing area is enclosed in a dashed box labeled 10.

- Data Bus:** A vertical bus at the top, labeled "データベース" (Data Base) and "20", connects to the external world.
- Program Memory:** A block labeled "プログラムメモリ" (Program Memory) is connected to the data bus via a bidirectional arrow.
- Internal Blocks:**
 - 命令デコーダ (Command Decoder):** Labeled 11, it receives "命令コード" (Command Code) from the latch circuit.
 - ラッチ回路 (Latch Circuit):** Labeled 12, it receives data from the database and outputs command codes to the decoder and address signals to the address counter.
 - アドレス変換命令検知信号 (Address Conversion Command Detection Signal):** Generated by the latch circuit, it controls the address counter.
 - スタックレジスタ D/W (Stack Register D/W):** Labeled 16, it handles stack operations.
 - AND Gate:** Labeled 14, it combines the "ラッチ指示信号" (Latch Instruction Signal) and "読み込み指示信号" (Load Instruction Signal).
 - タイミング発生回路 (Timing Generation Circuit):** Labeled 13, it generates timing signals based on the "CLK 割込信号" (Clock Interrupt Signal).
- Signal Flow:**
 - The database provides data to the program memory and the latch circuit.
 - The latch circuit outputs command codes to the decoder and address signals to the address counter.
 - The address counter outputs "変更アドレス値" (Changed Address Value) to the stack register and "割込み検出信号" (Interrupt Detection Signal) to the AND gate.
 - The AND gate's output goes to the timing generation circuit.
 - The timing generation circuit outputs "タイミング信号" (Timing Signal) back to the latch circuit and "処理終了信号" (Processing End Signal) to the stack register.
 - The stack register also outputs "書き込み指示信号" (Write Instruction Signal) to the AND gate.

20	10	CPU
	11	命令デコーダ
	12	ラッチ回路
	13	タイミング信号発生回路
	14	アンドゲート
	15	書き込み回路
	16	スタックレジスタ
	20	プログラムメモリ

アドレス

メインプログラム

0 A1

1 A2

10 A3

11 A4

20 終了命令

終了

割り込み発生入力

アドレス

割り込みプログラム

100 B1

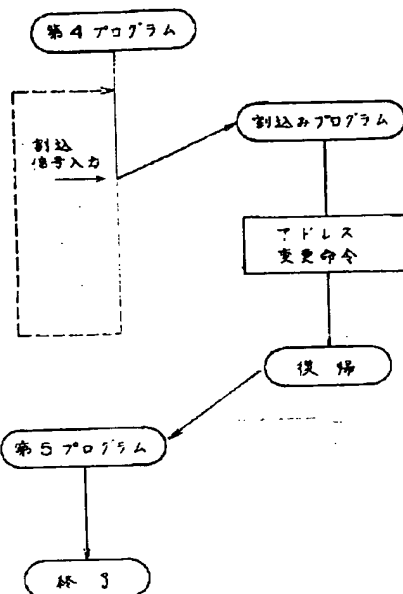
101 アドレス変更命令

102 E2

103 復帰命令

終了

【图 4】



【图 6】

```

graph TD
    S100([翻訳処理]) --> S101{読出し命令に  
アドレス変数命令  
?}
    S101 -- YES S110 --> S110[スタックポイン  
タ変数処理]
    S110 --> S102([終了])
    S101 -- NO S120 --> S120[その他処理]
    S120 --> S102
  
```

The flowchart for translation processing (S100) starts with a process box labeled '翻訳処理' (Translation Processing). It leads to a decision diamond labeled '読出し命令にアドレス変数命令?' (Is it an address variable instruction in the read instruction?). If the answer is 'YES' (labeled S110), it proceeds to a process box 'スタックポインタ変数処理' (Stack pointer variable processing). If the answer is 'NO' (labeled S120), it proceeds to a process box 'その他処理' (Other processing). Both paths eventually lead to an oval labeled '終了' (End).

